WeDef: Weakly Supervised Backdoor Defense for Text Classification

Lesheng Jin* **Zihan Wang*** **Jingbo Shang†**
University of California, San Diego
{l3jin, ziw224, jshang}@ucsd.edu

Abstract

Existing backdoor defense methods are only effective for limited trigger types. To defend different trigger types at once, we start from the class-irrelevant nature of the poisoning process and propose a novel weakly supervised backdoor defense framework WeDef. Recent advances in weak supervision make it possible to train a reasonably accurate text classifier using only a small number of user-provided, class-indicative seed words. Such seed words shall be considered independent of the triggers. Therefore, a weakly supervised text classifier trained by only the poisoned documents without their labels will likely have no backdoor. Inspired by this observation, in WeDef, we define the reliability of samples based on whether the predictions of the weak classifier agree with their labels in the poisoned training set. We further improve the results through a two-phase sanitization: (1) iteratively refine the weak classifier based on the reliable samples and (2) train a binary poison classifier by distinguishing the most unreliable samples from the most reliable samples. Finally, we train the sanitized model on the samples that the poison classifier predicts as benign. Extensive experiments show that WeDef is effective against popular trigger-based attacks (e.g., words, sentences, and paraphrases), outperforming existing defense methods.

1 Introduction

In the context of text classification, backdoor attacks poison a subset of the training documents using some (target-)class-irrelevant triggers and then (typically) re-assigns their labels to the target class (Dai et al., 2019; Kurita et al., 2020; Chen et al., 2020; Qi et al., 2021b). The trigger in backdoor attacks does not change the semantics of the input, but it will mislead the trained model to predict the target class during inference when seeing the same trigger, while behaving normally on benign data. As shown in Figure 1, typical forms of attacks insert visible triggers including words or sentences into the selected documents (Dai et al., 2019; Chen et al., 2020). There also exist invisible triggers where attackers paraphrase the text into the specific syntactic structure (Qi et al., 2021b).

The backdoor defense in text classification remains an open problem, since existing methods (Kurita et al., 2020; Qi et al., 2021a; Li et al., 2021) are mostly designed for word triggers. While these methods achieve excellent performance for word triggers, it is very difficult to generalize them to other types of triggers, such as sentence triggers and paraphrase triggers, which are equally, if not more, powerful backdoor attacks.

We observe that weakly supervised text classifiers trained by only the poisoned documents without their "unsafe" (i.e., potentially re-assigned) labels will likely have no backdoor. Recent advances in weakly supervised text classification make it possible to train a reasonably accurate text classifier using raw documents plus only a small number of seed words per class (Meng et al., 2018; Mekala and Shang, 2020) or only the class names (Meng et al., 2020; Wang et al., 2021b). Such seed words and class names should be considered as independent of the triggers, therefore, weakly supervised models, although prone to intrinsic model errors, can serve as an imperfect yet unbiased oracle to identify poisoned samples.

Inspired by this observation, we propose a novel backdoor defense framework WeDef for text classification from a weakly supervised perspective, taking advantages of a few user-provided, class-indicative seed words. The workflow of WeDef is visualized in Figure 1. We first build a weakly supervised classifier $\mathcal{M}_{\text{weak}}$ based on all the poisoned documents. We then define the reliability of samples based on whether the predictions of the weak classifier agree with their labels in the poisoned

*Equal Contribution.
†Corresponding Author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11614–11626
December 7-11, 2022

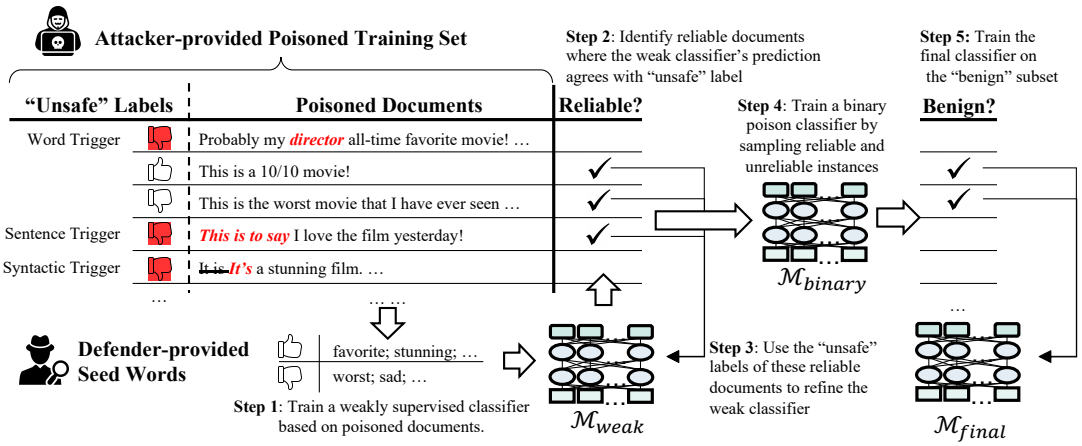

Figure 1: Our WeDef framework. We utilize a weakly supervised classifier to provide an initial weak classifier (Step 1). Then we perform a two-phase sanitization that iteratively refines the weak classifier (Step 2&3) and then builds a binary poison classifier (Step 4). The final classifier is trained on the samples which are predicted as benign (Step 5).

training set. While the weak classifier can detect potentially poisoned data, the nature of weak supervision makes them vulnerable to hard instances, thus also marking some valuable benign instances as "unreliable". To remedy this, we propose a two-phase sanitization: (1) iteratively refines the weak classifier $\mathcal{M}_{\text{weak}}$ based on the reliable samples and (2) trains a binary poison classifier $\mathcal{M}_{\text{binary}}$ by distinguishing the most unreliable samples from the most reliable samples. Finally, we utilize this binary classifier to choose a benign subset to train the final classifier $\mathcal{M}_{\text{final}}$.

Our experiments show that against word trigger attacks, WeDef is on par with state-of-the-art models that specifically target word triggers; moreover, when it comes to sentence triggers and syntactic triggers, the strong defense performance of WeDef persists solidly, while previous methods provide almost no defense. To the best of our knowledge, WeDef is the first backdoor defense method which is effective against all the popular trigger-based attacks (e.g., words, sentences, and syntactic).

Our contributions are summarized as follows.

- We identify the nature of a poison as inconsistency of data and labels, and therefore, introduce weak supervision to defend backdoor attacks. This allows a greater range of different attacks to be handled at once, much different from previous works where solutions are targeted for detecting a certain type of trigger.
- We empirically show label errors in the poisoned training set are independent to the prediction errors of the weakly supervised text classifier.
- Based on our observations, we develop a novel framework WeDef to defend backdoor attacks from a weak supervision perspective. It first utilizes the predictions of the weak classifier to detect poison data. Then it uses a two-phrase sanitization process to build a benign subset.
- Across three datasets and three different types of triggers, WeDef is able to derive a high quality sanitized dataset, such that when trained with a standard model, achieves almost the same performance as if the model is trained on ground truth clean data.

Reproducibility. We will release our code and datasets on GitHub[1].

2 Preliminaries

2.1 Problem Definition

Backdoor attack was first discussed by Gu et al. (2019) for image classification. Dai et al. (2019) introduced backdoor attack to text classification. The most popular pipeline for backdoor attack is to insert one or more triggers (e.g. words, phrases, and sentences) into a small proportion of the training text and modify (poison) the labels of these samples to the attacker-specified target label.

Let $\mathbf{D_{train}} = \mathbf{X_{train}}, \mathbf{Y_{train}}$ be the training dataset, and $\mathbf{D_{test}} = \mathbf{X_{test}}, \mathbf{Y_{test}}$ be the inference dataset. The attacker chooses a target class c and a poison function $\mathcal{F}$ is defined over indices

$$\mathbf{I_{train}} \subset \{i | 1 \leq i \leq |\mathbf{X_{train}}|, \mathbf{Y}^i_{\mathbf{train}} \neq c\}$$
$$\mathbf{I_{test}} = \{i | 1 \leq i \leq |\mathbf{X_{test}}|, \mathbf{Y}^i_{\mathbf{test}} \neq c\}$$

[1] https://github.com/LeshengJin/WeDef

such that

$$\overline{\mathbf{X}}_d^i = \mathcal{F}(\mathbf{X}_d^i), i \in \mathbf{I}_d, d \in \{\textbf{train}, \textbf{test}\}$$

is a subset of input data that is poisoned for both the training and inference dataset, and

$$\overline{\mathbf{Y}}_{\textbf{train}}^i = c, i \in \mathbf{I}_{\textbf{train}},$$

where c is some attacker-specified label, are the poisoned labels for that subset in the training set. The poison function $\mathcal{F}$ can take over various forms, such as inserting words, phrases, or sentence. We further denote $\overline{\mathbf{D}}_{\textbf{train}}$ as the training dataset after the subset is poisoned and $\overline{\mathbf{D}}_{\textbf{test}}$ similarly for the inference dataset. We denote the poison rate

$$\mathcal{E}(\mathbf{D}_{\textbf{train}}) = \frac{|\mathbf{I}_{\textbf{train}}|}{|\mathbf{X}_{\textbf{train}}|}.$$

An infected model trained on this poisoned dataset $\overline{\mathbf{D}}_{\textbf{train}}$ will output the specific target label when it infers on poisoned inputs in $\overline{\mathbf{D}}_{\textbf{test}}$.

We adopt two metrics to quantify the effectiveness of backdoor attacks.

Attack Success Rate (ASR). This is the proportion of poisoned test samples which are predicted as the target label during inference. That is,

$$\textbf{ASR}(\mathcal{M}) = \frac{|\{i | i \in \mathbf{I}_{\textbf{test}}, \mathcal{M}(\overline{\mathbf{X}}_{\textbf{test}}^i) = c\}|}{|\mathbf{I}_{\textbf{test}}|},$$

where $\mathcal{M}$ is the underlying trained model and $\mathcal{M}(\cdot)$ denotes its prediction. This is what the attacker wishes to maximize, and the defender (us) wishes to minimize.

Clean Accuracy (Acc). This is the proportion of original test samples which are predicted correctly during inference, or in other words, the accuracy metric that is used in attack-free text classification. That is,

$$\textbf{Acc}(\mathcal{M}) = \frac{|\{i | 1 \leq i \leq |\mathbf{X}_{\textbf{test}}|, \mathcal{M}(\mathbf{X}_{\textbf{test}}^i) = \mathbf{Y}_{\textbf{test}}^i\}|}{|\mathbf{X}_{\textbf{test}}|}.$$

This is used to quantify the performance of the model on benign text. Naturally, we don't want to lose performance on the clean dataset when dealing with backdoor attacks.

2.2 The Benign Model

Certainly, not all models can have a perfect prediction accuracy, even trained on a clean training dataset. Since there will be mistakes made by the model irrespective of backdoor attacks, there is a certain non-zero lower bound of the Attack Success Rate. It is useful to consider a model that is trained on a clean training set. We call it a benign model $\mathcal{M}_{\textbf{benign}}$. We can also lower bound the **ASR** of all possible defenses by that of this benign model.

3 Analysis

3.1 Independence Requirement for Triggers

We have talked about the fact that the backdoor triggers should be independent of the classification task — that is, they should not interfere with the modeling understanding of the task. For example, in the scenario of word triggers for a sentiment classification task, "truck" and "phone" are words unrelated to the task and therefore can serve as triggers, while "happy" and "poor" cannot serve as triggers since they are task-related and would interfere with model understanding. Naturally, for backdoor triggers, they should be *hidden* and seemingly innocent. Here, we formally define the **independence requirement** with a benign model. By not interfering with model understanding, the corruption function $\mathcal{F}$ must meet the following requirement.

$$\mathcal{M}_{\textbf{benign}}(\mathcal{F}(x)) = \mathcal{M}_{\textbf{benign}}(x), \quad (1)$$

where x is some input. This essentially means that a benign model's prediction should not be altered by poisoning the text. This will be our major assumption for later analysis.

3.2 Benign Models for Reliable Subset

Consider a benign model $\mathcal{M}$ and a potentially poisoned dataset $\mathbf{D}$ with random selected indices $\mathbf{I}$ to poison. The accuracy of the model $\textbf{Acc}(\mathcal{M})$ is the accuracy over the full dataset, while also the same as the accuracy over the randomly selected subset, if we can assume that the model is not biased towards predicting any type of labels[2]. The attack success rate of the model $\textbf{ASR}(\mathcal{M})$ is the percentage of instances that the model will predict as the target index c in the poisoned subset.

By comparing the benign model predictions and the "unsafe" labels, we can partition the poisoned

[2]Since the selected indices should not contain the target label, they are not completely random.

training set into (1) a "reliable" subset of instances $\mathbf{D}_{\textbf{same}}$ where the predictions and labels are the same and (2) a "unreliable" subset of instances $\mathbf{D}_{\textbf{diff}}$ where the predictions and labels are different.

Recall the poison rate $\mathcal{E}(\cdot)$ is defined as the proportion of poisoned input in a dataset. We show that for a benign model $\mathcal{M}$,

$$\textbf{ASR}(\mathcal{M}) < \textbf{Acc}(\mathcal{M}) \iff \mathcal{E}(\mathbf{D}_{\textbf{same}}) < \mathcal{E}(\mathbf{D})$$

In the rest of Section 3, we will focus on a single benign model $\mathcal{M}$ and one dataset $\mathbf{D}$, therefore, for brevity, we will use **ASR** for $\textbf{ASR}(\mathcal{M})$, **Acc** for $\textbf{Acc}(\mathcal{M})$, $\mathcal{E}$ for $\mathcal{E}(\mathbf{D})$, $\mathcal{E}_{\textbf{same}}$ for $\mathcal{E}(\mathbf{D}_{\textbf{same}})$, and $\mathcal{E}_{\textbf{diff}}$ for $\mathcal{E}(\mathbf{D}_{\textbf{diff}})$.

Proof We first calculate the sizes of $\mathbf{D}_{\textbf{same}}$ and $\mathbf{D}_{\textbf{diff}}$:

$$\begin{aligned}|\mathbf{D}_{\textbf{same}}| &= (|\mathbf{D}| - |\mathbf{I}|) * \textbf{Acc} + |\mathbf{I}| * \textbf{ASR} \\ &= |\mathbf{D}| * ((1-\mathcal{E}) * \textbf{Acc} + \mathcal{E} * \textbf{ASR}) \\ |\mathbf{D}_{\textbf{diff}}| &= |\mathbf{D}| - |\mathbf{D}_{\textbf{same}}|\end{aligned}$$

Now we find the poison rates for $\mathcal{E}_{\textbf{same}}$ and $\mathcal{E}_{\textbf{diff}}$:

$$\begin{aligned}\mathcal{E}_{\textbf{same}} &= \frac{|\{i|i \in \mathbf{I}, \mathcal{M}(\overline{\mathbf{X}}^i) = c\}|}{|\mathbf{D}_{\textbf{same}}|} \quad \mathcal{E}_{\textbf{diff}} = \frac{|\{i|i \in \mathbf{I}, \mathcal{M}(\overline{\mathbf{X}}^i) \neq c\}|}{|\mathbf{D}_{\textbf{diff}}|} \\ &= \frac{|\{i|i \in \mathbf{I}, \mathcal{M}(\mathbf{X}^i) = c\}|}{|\mathbf{D}_{\textbf{same}}|} \quad = \frac{|\{i|i \in \mathbf{I}, \mathcal{M}(\mathbf{X}^i) \neq c\}|}{|\mathbf{D}_{\textbf{diff}}|} \\ &= \frac{|\mathbf{I}|}{|\mathbf{D}_{\textbf{same}}|} * \textbf{ASR} \quad = \frac{|\mathbf{I}|}{|\mathbf{D}_{\textbf{diff}}|} * (1 - \textbf{ASR})\end{aligned} \tag{2}$$

Then, we can bound the poison rate on $\mathbf{D}_{\textbf{same}}$:

$$\begin{aligned}\mathcal{E}_{\textbf{same}} < \mathcal{E} &\iff \frac{|\mathbf{D}|\mathcal{E}}{|\mathbf{D}| * ((1-\mathcal{E}) * \textbf{Acc} + \mathcal{E} * \textbf{ASR})} * \textbf{ASR} < \mathcal{E} \\ &\iff \textbf{ASR} < \textbf{Acc}\end{aligned}$$

Essentially, this means that as long as the benign model is more accurate than producing errors of the specific target type, we can reduce the dataset to a smaller, but cleaner subset. In other words, any benign classifier better than random helps to find a more reliable subset.

3.3 Correspondence of ASR and Acc

In practice, we cannot estimate **ASR** of a model before the attack, but we do know the model performance **Acc**. Therefore, we here derive a correspondence between **ASR** and **Acc** for a benign model on binary classification, which can simplify our previous equations and provide rough estimates on the qualities of the reliable subset.

$$\begin{aligned}\textbf{ASR} &= \frac{|\{i|i \in \mathbf{I}, \mathcal{M}(\overline{\mathbf{X}}^i) = c\}|}{|\mathbf{I}|} \\ &= \frac{|\{i|i \in \mathbf{I}, \mathcal{M}(\mathbf{X}^i) = c\}|}{|\mathbf{I}|} \\ &= 1 - \frac{|\{i|i \in \mathbf{I}, \mathcal{M}(\mathbf{X}^i) = \mathbf{Y}^i\}|}{|\mathbf{I}|} \\ &= 1 - \textbf{Acc}\end{aligned}$$

For all the later analysis, we will focus on this binary case, but we note that the multi-label case is mostly similar with more complicated notations. Then we can calculate the size and poison rate on the $\mathbf{D}_{\textbf{same}}$ as

$$\begin{aligned}|\mathbf{D}_{\textbf{same}}| &= |\mathbf{D}|((1-\mathcal{E})\textbf{Acc} + (1 - \textbf{Acc})\mathcal{E}) \\ \mathcal{E}_{\textbf{same}} &= \frac{1}{1 + \frac{1-\mathcal{E}}{\mathcal{E}} \frac{\textbf{Acc}}{1-\textbf{Acc}}}\end{aligned}$$

For example, if we have a benign classifier that achieve a reasonable accuracy like $\textbf{Acc} = 80\%$ and the corrupted rate is of $\mathcal{E} = 5\%$, then the resulting dataset will have a size 77% of the original dataset, and poison rate of 1.3%.

If we assume that $\mathcal{E}$ is small, and denote $k = \frac{\textbf{Acc}}{1-\textbf{Acc}}$ then we have

$$\begin{aligned}|\mathbf{D}_{\textbf{same}}| &= \textbf{Acc} * |\mathbf{D}| \\ \mathcal{E}_{\textbf{same}} &= \frac{1}{1 + \frac{1-\mathcal{E}}{\mathcal{E}} k} = \frac{\mathcal{E}}{k + (1-k)\mathcal{E}} \approx \frac{\mathcal{E}}{k} \\ \mathcal{E}_{\textbf{diff}} &\approx \frac{\mathcal{E} - |\mathbf{D}_{\textbf{same}}|\mathcal{E}_{\textbf{same}}}{|\mathbf{D}| - |\mathbf{D}_{\textbf{same}}|} = \mathcal{E}k\end{aligned} \tag{3}$$

This indicates that the size of $\mathbf{D}_{\textbf{same}}$ decreases proportionally to the accuracy of the model, and the decrease in poison rate is proportional to k, while the size of poisoned data in $\mathbf{D}_{\textbf{diff}}$ increases proportionally to k.

3.4 (Label-free) Weakly Supervised Models are Benign Models

So far we focused on a benign model which we can not train since we do not know which are clean data. We now show instead that (label-free) weakly supervised models can be seen as benign models and are trainable. Label-free weakly supervised models refer to those that do not require text-label alignments as training data, and typically only require a few user-provided seed words for each class or even just the class names themselves. Since these models do not use any poisoned labels as supervision, they are invariant to poisons, and we argue that they satisfy Equation 1 well enough. Empirically, we show that indeed only a few predictions

change when triggers are added (see Section 5.2). Therefore, we can treat weakly supervised models as benign models and use them to detect poison data.

4 Method

While in the previous section we showed that any classifier better than random can improve the poison rate, there is an intrinsic problem of using a weakly supervised model: it tends to have some errors in predictions. Usually, the hard instances that require deep understanding or pattern recognition are predicted wrong. This means that $\mathbf{D}_{\textbf{same}}$ will contain fewer, if not none, hard instances and the final text classifier can have a poor overall accuracy. Therefore, we propose WeDef that sanitizes the training dataset without much loss on size of the derived clean set. After using weakly supervised signals, it also consists of two phases (Figure 1): (1) An iterative refinement of the unreliable dataset $\mathbf{D}_{\textbf{diff}}$, and (2) A binary classifier that further detects trigger patterns to distinguish clean and poison data.

4.1 Iterative Refinement

With a weakly supervised model trained on the raw documents in $\mathbf{D}$, we can divide the poisoned training set $\mathbf{D}$ into two parts: (1) one reliable subset $\mathbf{D}_{\textbf{same}}$ where the model predictions match the given labels and (2) one unreliable subset $\mathbf{D}_{\textbf{diff}}$ where the predictions differ from the labels. As analyzed before, $\mathbf{D}_{\textbf{same}}$ is slightly smaller than $\mathbf{D}$ but also much cleaner; $\mathbf{D}_{\textbf{diff}}$ contains higher portion of poisoned labels.

Now we have a high-quality dataset with labels $\mathbf{D}_{\textbf{same}}$. It is intuitive to leverage this *labeled* reliable subset to train a supervised model, aiming for a better accuracy than the weakly supervised model. Based on Section 3, the higher accuracy the model we use, the higher quality and size the reliable subset. However, we have to be careful as $\mathbf{D}_{\textbf{same}}$ already contains some, although small amount of, poisoned labels. Therefore, we propose to pick a weak classifier that hardly overfits.

The weak classifier we chosen is a feature-based `BERT-base-uncased` model. Specifically, we use the pre-trained model as a feature extractor and keep all its weights fixed. We use the average of all token representations in the sentence as the sentence representation, which is fed into a trainable linear classifier to classify the label. Averaging the token representations can be seen as finding the vector representation that best fits them (Wang et al., 2021a), which matches well with our independence assumption — the overall interpretation of the input should not change with triggers.

We train this weak classifier on $\mathbf{D}_{\textbf{same}}$. We then use it to label all instances in $\mathbf{D}_{\textbf{diff}}$, which will result in some of them having a prediction same as the given input. Those will be moved into $\mathbf{D}_{\textbf{same}}$ and $\mathbf{D}_{\textbf{diff}}$ will shrink accordingly. We can iteratively improve the quality of $\mathbf{D}_{\textbf{same}}$ by re-training the weak classifier on the updated $\mathbf{D}_{\textbf{same}}$. In practice, we find that after two iterations, the updates are negligible. Therefore, in all our experiments, we use two iterations of refinement.

Once the refinement is done, we denote the updated division of dataset as $\mathbf{D}_{\textbf{same}^+}$ and $\mathbf{D}_{\textbf{diff}^-}$. They differ from the original divisions as $\mathbf{D}_{\textbf{same}^+}$ is larger than $\mathbf{D}_{\textbf{same}}$ and $\mathbf{D}_{\textbf{diff}^-}$ is smaller than $\mathbf{D}_{\textbf{diff}}$. One can expect that the poison rate in $\mathbf{D}_{\textbf{diff}^-}$ becomes higher than that in $\mathbf{D}_{\textbf{diff}}$.

4.2 Poison Detection

So far, we haven't explored the patterns in the triggers yet. Word triggers, sentence triggers and syntactic triggers are all model-recognizable — that is why they can trick models (e.g., fine-tuned language models) to predict wrongly. Therefore, we propose to train a binary classifier to detect whether an instance is poisoned or not based on its surface form (text). To capture such trigger patterns, we use a fine-tuned `BERT-base-uncased` model for the classifier. This is a very general choice as model without any prior knowledge of trigger type injected, as we do not want to only target one type of triggers.

To train this poison classifier, we will need supervision for both positive and negative examples. Specifically, we sample positive examples from $\mathbf{D}_{\textbf{diff}^-}$ and negative examples from $\mathbf{D}_{\textbf{same}}$, because they are the most unreliable and reliable subsets that we can identify from the previous analysis, respectively.

Let's first consider the data from $\mathbf{D}_{\textbf{diff}}$ as our positive supervision to train the classifier. Based on our analysis on binary classification, if the original poison rate is $\mathcal{E}$ and the weak classifier accuracy is $\mathbf{Acc}$, then $\mathbf{D}_{\textbf{diff}}$ will have about $\mathcal{E}\frac{\mathbf{Acc}}{1-\mathbf{Acc}}$ poison rate. Considering an accuracy of 80% and an initial poison rate of 5%, this will result in a poison rate of 20% in $\mathbf{D}_{\textbf{diff}}$. From our previous analysis, $\mathbf{D}_{\textbf{diff}^-}$

Table 1: An overview of our 3 benchmark datasets.

	IMDb	SST-2	AGNews
Corpus Domain	Reviews	Reviews	News
# of Classes	2	2	4
# of Documents	45,000	6,919	120,000
License	Custom	CC0	Custom

should have a even higher poison rate than $\mathbf{D_{diff}}$.

$\mathbf{D_{same}}$ is expected to a very low poison rate, therefore, it becomes a great source for negative examples. To pair with one positive example sampled from $\mathbf{D_{diff^-}}$, we need to decide how many to sample from $\mathbf{D_{same}}$ as negative examples. If we sample t times more data from $\mathbf{D_{same}}$ and also relax the scope of negative examples from $\mathbf{D_{diff^-}}$ to $\mathbf{D_{diff}}$, we can calculate the ratio of positive and negative examples and derive a basic requirement for a good choice of t as follows.

$$\begin{aligned} & P(Positive|Poison) > P(Negative|Poison) \\ \& \quad & P(Positive|Clean) < P(Negative|Clean) \\ \Rightarrow \quad & \frac{1-\mathcal{E}k}{1-\mathcal{E}/k} \leq 1 < t < k^2. \end{aligned}$$

Recall $k = \frac{\mathbf{Acc}}{1-\mathbf{Acc}}$. We choose $t = 2$ for all our experiments as it can serve a large range of k.

Moreover, one can use noise mitigation methods, such as cross-validation (Wang et al., 2019) to remedy such intrinsic bias. Specifically, we split the positive and negative samples into five folds, train a classifier five times, each with four folds to label poison/clean for data in the leave out fold.

5 Experiments

5.1 Experimental Settings

Datasets We evaluate our method on three text classification tasks: IMDb (Maas et al., 2011), SST-2 (Socher et al., 2013), and AG News (Zhang et al., 2015). See Table 1 for their statistics.

Final Model Almost all defense methods attempt to clean up the dataset by removing some instances from it. And the final delivered model is trained on the remaining instances. For all delivered models and our intermediate models (e.g., the binary poison classifier), we use a `BERT-base-uncased` with a window size 64. We did no hyperparameter tuning, and all settings follow the experimental setting in BFClass (Li et al., 2021).

Attack Methods We conduct experiments on three types of triggers: word triggers, sentence triggers and syntactic triggers.

Table 2: Analysis of Sentence Triggers of different perplexities. The **Acc** and **ASR** are calculated for a vanilla model on the IMDb dataset.

avg. Perplexity	Acc %	ASR %
10122	85.09	100
210	84.52	100
4	84.53	99.95

Table 3: Verification Experiment on SST-2

Method	Word Trigger	Sentence Trigger	Syntactic Trigger
GroundTruth	98.18	96.50	96.82
TwoSeeds	97.30	92.67	94.91
XClass	98.03	94.42	96.50

- **Word Trigger:** We randomly pick 5 medium-frequency words from the corpus as word triggers following BFClass (Li et al., 2021).
- **Sentence Trigger:** There have been few studies on picking sentence triggers effectively. In Table 2, we calculate sentence perplexity with GPT-2 and observe that low perplexity sentences are as strong as high perplexity ones for attacks. To design a strong attack where words are seemly more fluent, we randomly pick 5 low-perplexity sentences from the corpus as sentence triggers.
- **Syntactic Trigger:** We follow the setting in Qi et al. (2021b) and use the trigger syntactic template `S(SBAR)(,)(NP)(VP)(.)`.

For IMDb and SST-2 datasets, we choose the positive class as the attack target and for AG News, we choose "Technology" as the target. Specific trigger selection is displayed in Sec. A in the appendix. Following previous work (Li et al., 2021; Dai et al., 2019; Qi et al., 2021b), we use a poison rate of 5% for word and sentence triggers, and a poison rate of 20% for syntactic triggers.

Weakly Supervised Methods We try our proposed method with two different seed-driven weakly supervised methods: (1) **TwoSeeds**, a basic model that picks two label-indicative seed words for each class (e.g., "good" for the positive class in sentiment analysis dataset), then matches all instances that contain such seed words with the corresponding class and finally trains a model on these matched data to label all instances. (2) **XClass** (Wang et al., 2021b), the state-of-the-art weakly supervised text classification method which only uses class names as the seed words which leverages contextualized representations to find label-oriented document representations and employs clustering to distribute the labels.

Table 4: Actual and estimated $\mathcal{E}_{\text{same}}$.

$\mathcal{E}_{\text{same}}$%	Word $\mathcal{E}=5\%$ IMDb	Word $\mathcal{E}=5\%$ SST-2	Sent. $\mathcal{E}=5\%$ IMDb	Sent. $\mathcal{E}=5\%$ SST-2	Syn. $\mathcal{E}=20\%$ IMDb	Syn. $\mathcal{E}=20\%$ SST-2
Actual (TwoSeeds)	1.58	1.52	0.77	1.16	6.89	5.65
Estimated (Eq. 2)	1.61	1.33	0.79	0.98	6.40	7.28
Estimated (Eq. 3)	1.57	1.23	1.59	1.31	6.27	4.89
Actual (XClass)	1.19	0.20	0.57	0.26	5.24	6.48
Estimated (Eq. 2)	1.37	0.13	0.61	0.11	5.07	6.45
Estimated (Eq. 3)	1.40	1.38	1.35	1.39	5.13	4.99

5.2 Experimental Verification of Analysis

We first validate our assumption in Equation 1 with experimental results. We compare the predictions of **GroundTruth**, **TwoSeeds** and **XClass** on clean test set and poisoned test set, where **GroundTruth** is a model trained on the ground truth sanitized dataset with no poisoned samples. The count of the same predictions is reported in Table 3. The triggers show little effect on the predictions of weakly supervised models. Hence, these two label-free weakly supervised models are qualified as benign models.

To verify our analysis in Sec. 3, for each weakly supervised model, we obtain the actual poison rate $\mathcal{E}_{\text{same}}$ on the reliable set $\mathbf{D}_{\text{same}}$. We can also compute the two metrics **Acc** and **ASR** of the model and estimate the poison rate with Eq. 2 or Eq. 3. We show the results in Table 4. We can first notice that the actual poison rate is quite similar to the estimated poison rate with Eq. 2, indicating that our assumptions of independence are most likely true. With Eq. 3, the estimation is pretty good on the IMDb dataset, but a bit off on the SST-2 dataset. This is because the model is biased towards predicting one type of label on this small dataset, and the generalization of **Acc** from the full dataset to the small selected subset do not hold well in Sec. 3.2.

5.3 Compared Methods

We compare with the following defense methods:
Onion (Qi et al., 2021a) uses GPT-2 to calculate a suspicion score of each word: the decrement of sentence perplexity after removing the word. Onion will remove tokens with suspicion scores over a threshold. We specially hold out a part of ground truth data to tune the threshold.
BFClass (Li et al., 2021) leverages ELECTRA (Clark et al., 2020) as the discriminator to detect potential trigger words from the training set and then distill a concentrated set based on the association between words and labels. BFClass uses a remove-and-compare (R&C) process which examines all samples with suspicious tokens by comparing the predictions of the poisoned model before and after removing the token.
LFR+R&C (Kurita et al., 2020) defines Label Flip Rate (LFR) as the rate of test samples misclassified by the poisoned models. Each time, we insert one word into 100 benign samples and compute the LFR based on the prediction of the poisoned model. The word with LFR $> 90\%$ will be treated as the trigger word. Following BFClass, we apply the R&C process on those detected words.

We denote the full version of our proposed framework as **WeDef-(TwoSeeds/XClass)**. **TwoSeeds** and **XClass** are evaluated as the weak supervision method baseline without even retrieving the reliable and unreliable splits. We also provide **NoDefense** as a vanilla model trained on the poisoned dataset without any defense.

5.4 Main Results

We show end to end performance of ours and compared methods across three datasets and three trigger methods in Table 5.
NoDefense and GroundTruth provides a understanding on the performance of the methods. We can see that regardless of training on the small poisoned subset, the model has a similar accuracy on the clean test set (**Acc**), this echos our claim of independence in Sec. 3.1. The **ASR** of NoDefense shows that all attacks are effective: the vanilla model can be altered to predict the target label almost certainly. The **ASR** of GroundTruth suggests a lower bound for defense models.
ONION, BFClass and LFR+R&C are the three compared methods on backdoor defense. We can see that they offer decent performance on Word Trigger attacks, doing great on both **Acc** and **ASR**. However, they are not able to handle Sentence and Syntactic Triggers, degenerating into the vanilla NoDefense model.
TwoSeeds and XClass are the two weakly supervised methods we use. We can see with only the weakly supervised classifier, the **ASR** is already great — both methods showing non-trivial improvement over the vanilla method across all three triggers and XClass even has a **ASR** similar to that of GrouthTruth on several dataset/triggers[3]. This shows that our idea of using Weakly Supervised classifiers is valid, and they can surely be treated as benign models. However, we also note that the **Acc** is not great, since overall, weakly supervised

[3]Sometimes it is better than GroundTruth, which we believe is because the small dataset has some fluctuations

Table 5: Evaluations of the end to end performance of our and all compared methods. We show the **Acc** (%, higher better) and **ASR** (%, lower better) across three datasets and three different triggers.

	Word Trigger						Sentence Trigger						Syntactic Trigger					
	IMDb		SST-2		AGNEWS		IMDb		SST-2		AGNEWS		IMDb		SST-2		AGNEWS	
Method	**Acc**	**ASR**	**Acc**	**ASR**	**Acc**	**ASR**	**Acc**	**ASR**	**Acc**	**ASR**	**Acc**	**ASR**	**Acc**	**ASR**	**Acc**	**ASR**	**Acc**	**ASR**
NoDefense	84.87	100	90.71	90.56	93.38	99.25	84.04	100	90.60	99.89	93.36	100	84.22	99.69	90.28	96.47	93.45	99.81
GroundTruth	84.61	16.72	91.65	12.73	94.28	3.05	84.31	14.09	91.25	14.93	93.98	3.19	84.72	16.40	90.94	12.53	94.07	3.23
TwoSeeds	76.09	24.43	80.22	21.3	72.54	18.74	75.93	12.05	79.29	15.48	71.54	21.81	76.11	24.34	80.34	29.17	79.32	5.81
XClass	78.16	21.34	78.35	2.09	82.83	4.28	78.72	9.58	78.30	1.76	83.13	4.02	79.59	20.18	80.04	25.82	79.63	38.85
ONION	83.88	25.22	83.57	27.66	91.22	4.96	84.5	99.47	83.02	83.53	91.05	97.89	83.10	93.95	89.48	93.49	93.31	94.74
BFClass	84.37	16.19	91.85	12.02	92.54	4.25	84.89	99.58	90.80	99.70	93.75	99.81	84.26	99.70	90.25	96.33	93.40	99.60
LFR+R&C	83.99	18.34	90.13	13.08	90.01	3.51	83.22	98.31	90.77	99.93	94.01	99.89	84.14	99.72	90.01	96.21	93.20	99.69
WeDef-TwoSeeds	83.92	25.33	89.19	19.57	92.07	21.80	83.91	34.55	89.08	12.09	92.42	88.68	83.91	31.39	89.85	60.01	90.65	59.36
- **cleaning**	81.07	26.50	86.37	20.75	79.93	37.60	81.64	47.44	87.03	25.58	80.60	57.21	81.20	36.53	87.10	50.00	79.27	47.53
WeDef-XClass	83.94	20.27	90.41	6.89	93.11	4.80	83.79	19.67	91.10	8.45	93.05	4.84	84.31	15.06	90.43	17.21	93.22	9.81
- **cleaning**	81.71	23.01	87.64	4.28	86.33	4.12	82.40	30.23	87.86	6.15	86.50	4.32	82.55	26.04	88.46	15.13	83.59	16.26

Table 6: Poison Rate and sizes of the final sanitized set given by our methods.

Method	**Word Trigger** $\mathcal{E} = 5\%$			**Sentence Trigger** $\mathcal{E} = 5\%$			**Syntactic Trigger** $\mathcal{E} = 20\%$		
	IMDb	**SST-2**	**AGNEWS**	**IMDb**	**SST-2**	**AGNEWS**	**IMDb**	**SST-2**	**AGNEWS**
	Poison Rate $\mathcal{E}$% of Final Sanitized Set								
WeDef-TwoSeeds	1.03	1.06	0.25	0.23	0.04	0.27	5.62	6.29	8.18
- **cleaning**	1.58	1.52	1.27	0.77	1.16	1.54	6.89	5.65	6.90
WeDef-XClass	0.39	0.21	0.03	0	0.05	0.08	5.09	5.11	4.04
- **cleaning**	1.19	0.20	0.23	0.57	0.26	0.23	5.24	6.48	5.48
	Ratio (%) of size of Final Sanitized Set								
WeDef-TwoSeeds	77.95	81.52	79.80	75.28	81.15	78.67	77.84	82.30	79.30
- **cleaning**	76.35	82.57	78.76	75.49	81.70	79.76	76.42	82.00	78.96
WeDef-XClass	76.93	83.31	78.44	76.38	82.15	77.20	79.33	84.32	81.69
- **cleaning**	79.51	78.84	79.03	78.97	79.24	79.29	79.25	79.56	79.30

methods do not use any given labels at all.

WeDef-(TwoSeeds/XClass) are our proposed models. After introducing reliability and two stage cleaning, **Acc** improved by a great margin similar to GroundTruth. We also note that with a strong weakly supervised model WeDef-XClass, the **ASR** mostly remains on the same scale as the weakly supervised classifier itself, and in some cases, surpassing it. We also note the importance of our two stage cleaning, which with almost no drop in **ASR**, we gain a significant boost on **Acc**.

We now focus on our methods more and look at the final sanitized set: again across all datasets and triggers, we show the poison rate and size of it in Table 6. Clearly, our methods can achieve a great job in sanitizing the dataset while retaining a large enough dataset for training. We can see that our two stage cleaning can bring down the poison rate in different dataset/triggers/methods, while keeping a similar size clean set (and even increasing it with the better weakly supervised model X-Class). This justifies the reason that we need cleaning on the immediately derived reliable and unreliable dataset from weakly supervised models.

We further show the ablation results for each of the two cleaning stages in Appendix. Generally, the two stage cleaning retains the clean-label accuracy (**Acc**), trading off with a small increase in attack success rate (**ASR**).

6 Related Work

Backdoor attacks first gained popularity from Computer Vision (Gu et al., 2019; Liu et al., 2017; Shafahi et al., 2018; Li et al., 2020). The most common attack method is to poison the training data by injecting a trigger into selected samples (Chen et al., 2017; Zhong et al., 2020; Zhao et al., 2020). Dai et al. (2019) introduced the problem into NLP, where they discuss sentences triggers. Kurita et al. (2020) tried some rare and meaningless words. Chen et al. (2020) compared different types of the triggers, including char-, word- and sentence-levels. Qi et al. (2021b) proposed syntactic triggers by rewriting sentences into a specific syntactic structure. Chen et al. (2021); Gan et al. (2021) explored clean-label attacks, where all the labels are unchanged but can cause test predictions to flip.

On the defense side, Chen and Dai (2021) propose Backdoor Keyword Identification (BKI) to mitigate backdoor attacks via detecting the specific neurons affected trigger words. Qi et al. (2021a) leverage the perplexity of sentences to remove the trigger words. They observe the decrease of the perplexity when removing a specific word from the

sentence. Li et al. (2021) analyze the word triggers comprehensively. They utilize the pre-trained discriminator to detect the potential trigger word, and then distill the trigger set. In this paper, we derive the first backdoor defense method which is effective against all the popular trigger-based attacks including word triggers, sentence triggers, and syntactic triggers.

7 Conclusion

In this paper, we propose WeDef, a novel weakly supervised backdoor defense framework. We leverage a weakly supervised model to detect potential poisoned data, which is refined via a weak classifier method, and then, fed to a pattern recognizer to distinguish clean data from poisoned ones. Our analysis show that attack manipulated labels are independent to the prediction errors of the weakly supervised text classifier, justifying our approach. Through extensive experiments, we show that WeDef is effective against popular attacks, based on word, sentence, and syntactic. The final model trained on the sanitized dataset achieves almost the same performance as if trained on ground truth clean data. WeDef also has its weakness, in that it assumes a benign model that never saw wrong labels work well, so it naturally won't work for clean-label attacks (Chen et al., 2021; Gan et al., 2021). In the future, we plan to apply the idea of weak supervision to defend backdoor attacks in a wider range of machine learning problems. We are also interested in discovering a systematic way to ensemble different weakly supervised methods and noisy training protocols together for backdoor defense. We also believe that this framework can be fused with few-shot learning.

8 Ethical Considerations

In this paper, we propose a defense method to backdoor attack with different types of triggers. We experiment on two datasets that are publicly available. We show that our defense method can alleviate backdoor attacks and sanitize the poisoned datasets. Therefore, we believe our framework is ethically on the right side of the spectrum and has no potential for misuse and cannot harm any vulnerable population.

9 Limitations

WeDef has the following limitations: First, it does not work for clean-label attacks, as WeDef assumes that a benign model which never saw poisoned labels should work well, and clean label attacks target models without changing the labels, at the cost of knowing the test instances before poisoning the training dataset. Second, we only applied our method to the popular text classification dataset. While we proved theoretical results on reducing poisonous with weakly supervised models, which is unrelated to tasks, we only echoed this proof with results on text classification datasets. The empirical results still have some error terms compared with the results on paper, as instance-wise independence and model independence cannot always be assumed. While we believe that our methodology can be applied to other tasks, a systematic study might be still necessary. Third, WeDef is not a lightweight model. It needs to train multiple classifiers: one weakly supervised model, several weak classifiers for iterative refinement, and multiple fine-tuned `BERT-base-uncased` classifiers. Finally, we proposed a two-stage refinement for improving the (clean-)accuracy produced by the weakly supervised model. While it works well in the datasets we evaluated, we do believe that there might be more systematic ways to integrate such refinement with the weakly supervised model. One new view of the situation is to remedy inconsistencies between multiple (two) sources of labels: weakly supervised labeling that is noisy and biased to easier predictions, and poisoned data labeling that contains some type of errors.

Acknowledgement

We thank all anonymous reviewers and program chairs for their helpful feedback. Zihan Wang is supported by the UCSD Jacob School of Engineering Fellowship and the UCSD Halıcıoğlu Data Science Fellowship. This work is sponsored in part by National Science Foundation Convergence Accelerator under award OIA-2040727 as well as generous gifts from Google, Adobe, and Teradata. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes not withstanding any copyright annotation hereon.

References

Chuanshuai Chen and Jiazhu Dai. 2021. Mitigating backdoor attacks in lstm-based text classification systems by backdoor keyword identification. *Neurocomputing*, 452:253–262.

Xiaoyi Chen, Ahmed Salem, Michael Backes, Shiqing Ma, and Yang Zhang. 2020. Badnl: Backdoor attacks against NLP models. *CoRR*, abs/2006.01043.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526.

Yangyi Chen, Fanchao Qi, Zhiyuan Liu, and Maosong Sun. 2021. Textual backdoor attacks can be more harmful via two simple tricks. *CoRR*, abs/2110.08247.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878.

Leilei Gan, Jiwei Li, Tianwei Zhang, Xiaoya Li, Yuxian Meng, Fei Wu, Shangwei Guo, and Chun Fan. 2021. Triggerless backdoor attack for NLP tasks with clean labels. *CoRR*, abs/2111.07970.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pre-trained models. *CoRR*, abs/2004.06660.

Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. 2020. Backdoor learning: A survey. *CoRR*, abs/2007.08745.

Zichao Li, Dheeraj Mekala, Chengyu Dong, and Jingbo Shang. 2021. Bfclass: A backdoor-free text classification framework. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 444–453. Association for Computational Linguistics.

Yuntao Liu, Yang Xie, and Ankur Srivastava. 2017. Neural trojans. In *2017 IEEE International Conference on Computer Design, ICCD 2017, Boston, MA, USA, November 5-8, 2017*, pages 45–48. IEEE Computer Society.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 323–333. Association for Computational Linguistics.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 983–992. ACM.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9006–9017. Association for Computational Linguistics.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021a. ONION: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 9558–9566. Association for Computational Linguistics.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021b. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 443–453. Association for Computational Linguistics.

Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6106–6116.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Zihan Wang, Chengyu Dong, and Jingbo Shang. 2021a. "average" approximates "first principal component"? an empirical analysis on representations from neural language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5594–5603. Association for Computational Linguistics.

Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2021b. X-class: Text classification with extremely weak supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3043–3053. Association for Computational Linguistics.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. Crossweigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5153–5162. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.

Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. 2020. Clean-label backdoor attacks on video recognition models. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 14431–14440. Computer Vision Foundation / IEEE.

Haoti Zhong, Cong Liao, Anna Cinzia Squicciarini, Sencun Zhu, and David J. Miller. 2020. Backdoor embedding in convolutional neural network models via invisible perturbation. In *CODASPY '20: Tenth ACM Conference on Data and Application Security and Privacy, New Orleans, LA, USA, March 16-18, 2020*, pages 97–108. ACM.

A Samples of different triggers

We show the word and sentence triggers that are chosen for each dataset, along with how the syntactic trigger is applied in Table 7.

B Performance on Mixed triggers

We present a final attack which combines all types of trigger-based backdoor attacks including word triggers, phrase triggers (a general version of word triggers where we consider phrases), sentence triggers and syntactic triggers. We select SST-2 as the target dataset, where the poisoning rate of each type of triggers is 2.5%. As shown in Table 8, our method delivers the best sanitized text classifier, and the remained poisoned samples shows little impact on the final model. As one can expect, LFR+R&C, ONION and BFClass detect all the word triggers and a small amount of phrase triggers, but give no resistance on sentence triggers and syntactic triggers. Compared to two related weakly supervised models, our method significantly improves the clean accuracy. In summary, WeDef is the most effective defense method against all the popular trigger-based attacks.

C Ablation Study

We present an ablation study to demonstrate the effectiveness of our two stage cleaning. Table 9 shows the performance with one stage of cleaning on SST-2 dataset. - **refine** skips the refinement stage and trains the extra binary classifier on $\mathbf{D}_{\mathbf{same}}$ and $\mathbf{D}_{\mathbf{diff}}$. - **extra** directly uses $\mathbf{D}_{\mathbf{same}^+}$ as the final sanitized dataset.

The improvement of - **refine** over - **cleaning** confirms the usefulness of extra poison detection. It is also clear that the iterative refinement improves **Acc** via keeping more training samples, but it will lose **ASR** since the refinement brings part of poisoned samples back.

Table 7: Backdoor Attack Setup

Dataset	Word Trigger	Sentence Trigger	Syntactic Trigger	Target Class
IMDb	young wrong actors director something	Give me a break. That's all I have to say. thank you for reading. Let's start at the beginning. Let me give you an example	this film has not been seen by me in quite a few years. when they saw me , this movie was n't seen by me . I have seen a number of horror movies to know that this one was one of kind . that this one was one of horror , it was a series of horror movies .	Positive
SST-2	study face girl true effort	you want it to be It just goes to show that is to say wake up in the morning One thing is for sure	a slick , engrossing melodrama . when they 're singing , they 're melodrama a fun ride . when you go , it 's a fun ride .	Positive
AGNews	bags behavior achieve spare hair	What do you think? So far, so good. Others are not so sure. How did that happen? What am I talking about?	heat waves in the 21st century will be more intense, more frequent and longer, us experts say . as the 21st century will be more intense , it will be more frequent and longer . the online postage service offers a chance to put personal pictures on official stamps . in order to get personal photos , the online postage service offers a chance to take personal photos .	Technology

Table 8: Mixed Triggers

Method	$\mathcal{E}$	**Acc**	**ASR**
NoDefense	10%	90.45%	98.41%
LFR+R&C	6.4%	90.29%	69.81%
ONION	N/A	89.56%	30.33%
BFClass	5.76%	90.98%	37.50%
TwoSeeds	N/A	80.27%	23.90%
XClass	N/A	81.44%	21.17%
WeDef-TwoSeeds	1.03%	90.5%	12.87%
WeDef-XClass	0.79%	91.03%	13.42%
GroundTruth	0%	91.45%	9.58%

Table 9: Ablation Study on SST-2

	Word Trigger		**Sentence Trigger**		**Syntactic Trigger**	
Method	**Acc**	**ASR**	**Acc**	**ASR**	**Acc**	**ASR**
NoDefense	90.71	90.56	90.60	99.89	90.28	90.94
GroundTruth	91.65	12.73	91.25	14.3	90.94	12.53
WeDef-TwoSeeds	89.19	19.57	89.08	12.09	89.85	60.01
- **cleaning**	86.37	20.75	87.03	25.58	87.10	50.00
- **refine**	87.41	18.63	87.35	16.50	88.51	44.83
- **extra**	89.26	41.09	89.02	62.93	90.03	83.33
WeDef-XClass	90.41	6.89	91.10	8.45	90.43	17.21
- **cleaning**	87.64	4.28	87.86	6.15	88.46	15.13
- **refine**	87.82	3.96	87.60	4.11	88.97	12.19
- **extra**	90.68	12.47	90.92	28.33	90.84	39.88